A Computing Approach Using Probabilistic Neural Networks for Instantaneous Appraisal of Rear-End Crash Risk

Anurag Pande & Mohamed Abdel-Aty

Department of Civil and Environmental Engineering, University of Central Florida, Orlando, FL 32816-2450

Abstract: *Computing and information technology has significantly increased the capabilities to collect, store, and analyze freeway traffic surveillance data. The most common forms of such data are collected using the underground loop detectors. In the recent past the potential of using these data for identification of crash-prone conditions has been explored. In the present work, application of probabilistic neural networks (PNN) is explored to identify conditions prone to rear-end crashes on the freeway. PNN is a neural network implementation of the well-documented Bayesian classifier. In this research the rear-end crashes observed on the Interstate-4 corridor in Orlando FL are divided into two groups based on the average traffic speeds observed around the crash location prior to the crash occurrence. Using decision tree-based classification it was observed that although these two groups of crashes have comparable frequencies, traffic conditions belonging to one of the groups (characterized by a low-speed traffic regime) are comparatively rare on the freeways. Hence, if those conditions are encountered on the freeway in real time, then conditions may be considered prone to rear-end crashes. As conditions belonging to the other group of rear-end crashes (characterized by a medium-to-high speed regime) are more commonly observed on the freeway, PNN-based classification models are developed for this group. The rear-end crashes along with a sample of randomly selected noncrash cases were used to calibrate the classifiers. The output layer of the PNN models was modified to provide a measure of crash risk, instead of the binary classification based on an arbitrary threshold. A desirable threshold on this output may be established to separate crash-prone conditions from "normal" freeway traffic.*

1 INTRODUCTION

Rear-end crashes on freeways are generally associated with traffic speed and its variation (Golob and Recker, 2001). Their direct association with traffic conditions on the freeway makes them one of the more "predictable" types of crashes on the freeways. The current focus of traffic management research is on identifying conditions prone to crashes and devising proactive counter measures. In this regard, methods to identify conditions prone to rear-end crashes are critical because they are the most frequent type of crashes on freeways. Moreover, their impact on freeway operation is also quite profound because most of these crashes are observed when the facility experiences medium to heavy demand. Before the methods to avoid crashes may be devised, a reliable framework for identification of crash-prone conditions should be developed.

The primitive element of a proactive traffic management system would be reliable models separating crash-prone conditions from "normal" traffic conditions in real time. Most of the existing real-time crash "prediction" models available in the literature are generic in nature (e.g., Lee et al., 2002; Lee et al., 2003; Abdel-Aty et al., 2004), that is, a single generic model has been used to identify all crashes (such as rear-end, sideswipe, or angle).

The reason for the generic nature of existing real-time crash risk assessment models could be that crashes are rare events and until sufficient effort has been devoted to data collection and preparation, the sample size would not be sufficient for individual crash types. In this regard, a rear-end crash database assembled over a 5-year period (1999 through 2003) from the 36.25-mile corridor of Interstate-4 in Orlando metropolitan area is used for this study. It was demonstrated in one of our recent studies (Pande and Abdel-Aty, 2006) that the rear-end crashes on this freeway can be grouped into two distinct clusters: one, the crashes that occur under extended congestion and two, the crashes that occur with relatively free-flow conditions prevailing 5–10 minutes before the crash. It was also shown by Pande and Abdel-Aty (2006) that while the former group of rear-end crashes could be efficiently "predicted," further classification models are needed for identification of the second group of rear-end crashes. The present study focuses on a probabilistic neural network (PNN) based approach to identify the conditions prone to a later group of rear-end crashes from a sample of randomly selected freeway traffic data.

2 BACKGROUND

Madanat and Liu (1999) were one of the first researchers to explore the idea of proactive traffic management by focusing on enhancement of existing incident detection algorithms with likelihood of incidents (e.g., crashes and overheating vehicles). Lee et al. (2002, 2003) developed and refined log-linear models to "predict" crashes using crash precursors estimated from loop detector data. It was found that the coefficient of temporal variation in speed has a relatively longer term effect on crash potential than density while the effect of average variation of speed across adjacent lanes was found to be insignificant. A study by Oh et al. (2001) also showed the 5-minute standard deviation of speed value to be the best indicator of "disruptive" traffic flow leading to a crash as opposed to "normal" traffic flow. In our previous work (Abdel-Aty and Pande, 2005) PNN models were developed for separating traffic data recorded before historical crashes from noncrash data. The generic PNN model developed in that study achieved 72.5% crash identification and overall (crash and noncrash) classification accuracy of 62.5%. Crash identification, 72.5%, is in fact the true positive rate and is also referred to as "Recall" (Davis and Goadrich, 2006).

The major shortcoming of these studies was that the inferences were made based on a "one-size-fits-all" approach. Conditions preceding crashes are likely to differ by type of crash and therefore the approach toward proactive traffic management should be the type (of crash) specific in nature. The models estimating crash risk for specific types of crashes would also be benefi cial for developing more specific remedial measures to improve the safety situation on the freeway, for example, the application of variable speed limits for rear-end crashes or a temporary "no lane-changing" sign to avoid an impending sideswipe crash.

In this regard, Golob and Recker (2001) showed that some crash types are more common under certain traffic conditions. Later, Golob et al. (2004) developed a classification scheme by which traffic conditions measured through the loop detectors can be classified into groups that differ in terms of likelihood of crash of each type. As noncrash data were not used in these studies, their findings albeit insightful, are not applicable in the framework of a system separating "crash prone" conditions from "normal" traffic conditions in real time.

In this study, we try to overcome these shortcomings by examining traffic data from a series of loop detectors and explore their relationship with rear-end crashes. The choice of rear-end crashes was obvious due to their high frequency and significant impact on freeway operation. Available relevant data belonging to 1,620 rear-end crashes (from 1999 through 2003) have

been used with noncrash data that are collected from randomly chosen corridor locations over the 5-year period (also from 1999 through 2003). The random sampling of non-crash locations enables us to explore the impact of "off-line" factors (e.g, presence of ramps), along with real-time traffic parameters, on occurrence of a rear-end crash.

3 STUDY AREA AND AVAILABLE DATA

The 36.25-mile Interstate-4 (I-4) corridor under consideration has a total of 69 loop detector stations, spaced out at nearly half a mile. Each of these stations consists of dual loops and measures average speed, occupancy, and volume over a 30-second period on each of the three through travel lanes in both directions. The loop detector data were continuously transmitted and archived by the UCF data warehouse. The source of crash and geometric characteristics data for the freeway is *FDOT* (Florida Department of Transportation) intranet server.

There were 2,179 rear-end crashes reported in the study area during the 5- year period (from 1999 through 2003). The size of the sample used in this article, however, would reduce to 1,620 based on the availability of corresponding loop data. From the FDOT database we extracted information such as the report number, date, and milepost location for each crash. Scanned copies of individual crash reports were then used to extract the reported time of each crash. The DOT milepost location was used to determine the station nearest to crash location. This station was referred to as "station of crash." A binary variable "stationf," indicating whether this nearest station (i.e., station of crash) is upstream or downstream of the crash location, was also created based on this information.

A critical issue identified in some of the studies discussed above (e.g., Lee et al., 2002) was that of the accuracy of the reported time of crashes. Fortunately, there is an automated system in place in Florida that records the exact time when a crash is reported to the Police. According to Florida Highway Patrol (FHP) officials, due to widespread usage of mobile phones, the difference between time of crash occurrence and its reporting is minimal. It was also pointed out by local traffic management authorities that the reported time of the crash in accident reports is corroborated through the video surveillance system available on the freeway. This information indicated that the time obtained from the crash reports is in fact very close to the actual time of crash occurrence. The reported time of the crash obtained from individual crash reports has, therefore, been used in the analysis presented in this article.

The loop data used in this study were originally available in the form of 30-second averages. To filter out the noise in 30-second data, it was decided that 5-minute level data aggregated across all lanes would be used in this study (see Pande et al., 2005 for more details). For all crashes these data were obtained from five different stations around the crash location. These stations include station of crash (referred to as Station F), two stations preceding Station F in the upstream direction (Station D and E), and two stations following Station F in the downstream direction (Stations G and H). Hence, D would be the farthest station upstream and H would be the farthest station downstream. The arrangement of the stations with respect to crash location may be found in Pande and Abdel-Aty (2006). The information extracted from these stations included 5-minute averages and standard deviations of speed, volume, and lane-occupancy obtained using the raw 30-second data. The 5-minute time period between time of the crash and 5 minutes prior to the crash was named as time slice 1, while the interval between 5 and 10 minutes prior to the crash was referred to as time slice 2. The decision to examine data only up to 10 minutes prior to the crash was based on the results obtained by Pande (2005).

A four-letter nomenclature procedure has been used in this study for the variables. The first letter is "A" or "S" representing average or standard deviation, the second letter is "S," "V," or "O" representing speed, volume, or lane-occupancy, the third letter represents the station D, E, F, G, or H. The last letter in any traffic data-related variable represents time slice 1 or 2. The variable named "*SSD2*," for example, would represent the standard deviation of 30 speed observations during the 5-minute period of 5–10 minutes prior to a crash at station "D," which is

the farthest upstream station. Note that due to random intermittent failure of certain detectors, traffic data were only available for 1,620 (out of the total 2,179) rear-end crashes.

As mentioned previously, a random sample of non-crash cases have been used in the analysis. To generate a random noncrash sample, a 5-year period was divided into 2,629,4401-minute periods (60 minutes x 24 hours x 1,826 days over 5 years = 2,629,440 1-minute periods), which would be the number of options available to choose the "time of noncrash." Similarly, we have 138 stations (69 stations in two directions, eastbound and westbound) to choose as "station of noncrash." In all, we can choose from 362,862,720 (2,629,440 1 minute periods x 2 directions x 69 stations) options to draw a random combination of time, station, and direction to assign as a random noncrash case. 150,000 such combinations were selected randomly as the noncrash cases. Using the time and stations for noncrash cases, all traffic parameters extracted for crash cases were extracted for noncrash cases as well. Out of these 150,000 available random noncrash cases, a noncrash sample of appropriate size may be drawn depending on the sample size requirements of the methodology used for analysis.

After the assembly of traffic parameters geometric features of the freeway at the locations of aforementioned crash and noncrash cases were collected. The geometric design feature for the crash location was extracted based on the milepost of respective crashes (variable named "base milepost"). As random non-crash cases were extracted based on the "station of crash," the variable "base milepost" was not available for them. Therefore, it was decided to "assign" a milepost location to each random noncrash case. As the station of crash was available for each noncrash case, the milepost assigned to it was a random milepost generated from within the influence area of the station of crash. The influence area for any station was defined as the section between the midpoints of the station of interest and the stations up and downstream. Hence, for any point within the boundaries of the influence area corresponding to a particular station, that station would be the nearest loop detector station. To assign "base milepost" to random noncrash cases the mileposts corresponding to these boundaries were estimated for every loop detector station in the study area. These mileposts were merged with each noncrash case based on the station of the crash associated with it. A random number was then chosen between the milepost of these boundaries and assigned as "base milepost" for that noncrash case.

The milepost location of the ramps on the Interstate-4 corridor was known from the FDOT database. Using this information, along with the "base milepost" of each crash and noncrash cases, we created four continuous variables, namely, "upstreamon," "upstreamoff," "downstreamon," and "downstreamoff," indicating the distance of the nearest ramp of the respective type from crash location. Other geometric design features such as the curvature and number of lanes at the crash and noncrash locations were also collected based on the "base milepost." The database assembled herein now includes 5 years of crash (and noncrash) data for a 36.25 mile freeway corridor along with corresponding traffic information and geometric design features. It is by far the most comprehensive database assembled for developing crash risk assessment models.

4 PRELIMINARY DATA ANALYSIS 4.1 Analysis of traffic speed distributions

As part of the preliminary analysis frequency histograms for variables *ASD1, ASF1*, and *ASH1* over all rear-end crashes were examined. It was observed that all three histograms had the shape of two adjacent approximately mound-shaped distributions. The distribution for *ASD1* is depicted in Figure 1. Note that *ASD1* is the average of speeds measured from the three lanes at Station D (Station located approximately 1-mile upstream of the station of crash) during the 5-minute period leading to the crash (*Slice 1*). *ASF1* and *ASH1* are the same parameters measured at Station F and Station H, respectively. The shape of these distributions suggested that crashes belonging to the two adjacent approximately mound-shaped distributions need to be analyzed

separately. Therefore, the rear-end crashes were grouped into two distinct clusters based on the distributions of speeds at the aforementioned three stations. The first cluster consisted of crashes that occur under extended congestion on the freeway. The average speeds were relatively higher before the second cluster of crashes. The traffic speed conditions corresponding to the former group were called Regime 1 and those corresponding to the later were called Regime 2. Simple "if-then" rules consisting of average traffic speeds at the aforementioned stations during time slice 2 were formulated (based on classification tree methodology) to separate the traffic conditions belonging to the two regimes in real time. Note that speeds from time slice 2 (5–10 minutes before the crash) were proposed to be used for identifying the traffic regime in real time instead of time slice 1 (0–5 minutes before the crash) so that there is time available for data analysis and information dissemination in a real-time scenario. These rules and the complete procedure to formulate these rules may be found in Pande and Abdel-Aty (2006).

5 REAL-TIME IDENTIFICATION OF REAR-END CRASHES

By application of a classification of tree-based rules for separating two traffic regimes on a sample of randomly selected noncrash cases it was found that the Regime 1 conditions are very rare on the freeways (6.63% of all observations) while the Regime 2 conditions are much more commonly observed (93.37% of all observations). Based on application of these rules on the rear-end crashes, Regime 1 crashes make up about 46% of the crashes while 54% of crashes occurred under Regime 2 conditions.

The real-time identification strategy is essentially based on the observation that traffic conditions belonging to Regime 1 occur very infrequently (only 6% in the randomly selected loop data patterns) on freeways but make up close to 46% of rear-end crashes. Hence, it would be reasonable to classify every pattern that fits into the criterion of Regime 1 (low-speed conditions specified in Pande and Abdel-Aty, 2006) as a crash. This way we would identify 46% of rear-end crashes by issuing a warning only 6% to 7% of the time. Regime 2 crashes make up a bigger portion of rear-end crashes even as the corresponding traffic conditions are way more commonly observed on the freeway. Therefore, we need classification models that would provide a measure to separate the crash and noncrash cases among all Regime 2 traffic conditions. The focus of this study is to explore PNN as a method for assessing the risk of Regime 2 (medium-to-high speed conditions specified in Pande and Abdel-Aty, 2006) rear-end crashes. These models were calibrated/validated using 878 Regime 2 rear-end crashes and a sample of 4,972 random non-crash observations also belonging to Regime 2.

[Insert Figure 1]

5.1 Review of PNN

PNN is a neural network implementation of the well-established multivariate Bayesian classifier. It uses Parzen estimators to construct the probability density functions for competing classes (Specht, 1990, 1996). Parzen estimator relies on a weight function $W(d)$ (i.e., kernel) that has the largest value at $d = 0$. Its value decreases rapidly with the increase in absolute value of "d" (Masters, 1995). The probability density of existing observations around a new data point (x) is the scaled sum of Parzen estimators for all the existing observations. It may be represented by the following equation:

$$f(x) = \frac{1}{n\sigma} \sum_{i=1}^{n} W\left(\frac{x - x_i}{\sigma}\right) \qquad (1)$$

The scaling parameter σ (i.e, spread value) defines the width of the bell curve that surrounds each sample point. Too small values of σ will cause individual training cases to have too much

influence (thereby losing the benefit of aggregate information) while extremely large values will cause so much blurring that the details of density will be lost (Masters, 1995). To classify a new observation using PNN, pdf (probability density function) of the observations belonging to competing classes are estimated in the vicinity of the new observation. Whichever class has higher density in the vicinity of this new observation gets assigned to the new observation. Hence, PNN may be understood as kernel discriminant analysis implemented in the form of a network resembling the architecture and information flow of a neural network (Specht,1990).

Figure 2 depicts the neural network implementation for a 2-class classification problem involving *p*-dimensional input. The pattern layer contains one neuron for each training case while the summation layer has one neuron for each class. In the training phase, each training case (patterns with known classification) gets stored in a neuron of the pattern layer. To classify an unknown input pattern the execution starts by simultaneously presenting this input vector to all pattern layer neurons. Each pattern neuron then computes a Euclidean distance measure between the input vector and the training case represented by that neuron. It then subjects the distance measure to an exponential weight function similar to the one used in the Parzen estimator (i.e., $W(d)$).

The following layer contains two summation units each belonging to a single class (crash or noncrash for the present application). These summation units sum up the outputs from pattern layer units corresponding to that summation neuron's class. The attained activation of each of the two summation layer units is the estimated density function (multivariate form of Equation (1)) value for that population class in the vicinity of the unknown input pattern up for classification (Masters, 1995). To achieve binary classification the output layer may be used as a threshold discriminator. In fact, the PNN architecture used in our previous study (AbdelAty and Pande, 2005) provided binary output as crash or noncrash through the neuron in the output layer.
[Insert Figure 2]

In this study, however, the function of the output layer unit was modified because simply using classification accuracy results (based on a threshold discriminator function) can be misleading. It was also suggested by Provost et al. (1998) in their study on comparison of induction algorithms. The transfer function (from summation layer to output layer) was modified to the "soft-max" function. The softmax function estimates the exponential of the pattern layer neuron corresponding to the crash class normalized by the sum of exponentials for the two summation layer neurons (Ma and Morgan, 1995). In other words, it normalizes the attained activation of the summation layer neuron for the "crash" class between 0 and 1. Hence, the softmax function provides a continuous output similar to posterior probability and not an arbitrary "threshold"-based classification. The closer this modified output for an observation is to unity the more likely it is for that observation to be a crash. Also, note that using the output layer as a mere threshold discriminator is equivalent to using the posterior probability of 0.5 as the threshold and classifying all observations above 0.5 as a crash and below 0.5 as a noncrash. It reflects the performance of the model at a predetermined threshold on output posterior probability. Due to inherent imbalance (between crash and noncrash cases) in the training and validation sample, classification based on a predetermined threshold is inappropriate. For example, with 15% of crashes in the sample, classification accuracy as high as 85% could be achieved by a model that merely classifies every observation as a noncrash. Such a model would of course be useless for crash identification. One way to address this issue would be to use cost-sensitive "training" of the network. It would require establishing the cost of misclassifying a crash relative to misclassifying a noncrash. However, estimation of these costs is not a trivial issue. In this regard, posterior probability provides a continuous measure of the crash risk that may be used to assess traffic conditions relative to each other. The classification performance for models, if necessary, may be evaluated at different thresholds of posterior probability.

5.2 Calibration of PNN models for Regime 2 crashes

The PNN models for the Regime 2 rear-end crashes were developed in three stages. In the first stage traffic

parameters only from Station F were included as inputs along with the geometric design parameters. In the subsequent stages parameters from three (Station D, E, F) and five stations (Station D through H) were included as inputs. A classification tree was used for variable selection at all three stages (Breiman et al., 1984). Separate lists of significant variables used in each of the three stages are provided in Table 1. Note that due to the different set of variables used the dimensionality of the input vector (denoted as "*p*" in the previous section and Figure 2) would be different for each of the three stages.

Prior to the variable selection procedure transformations were applied to critical off-line factors, represented by variables "*base milepost*," "*upstreamon*," "*upstreamoff*," "*downstreamon*," "*downstreamoff*," and "*timeofcrash*." In their original form these variables were not suitable for real-time identification of crash-prone conditions because their value would change continuously through the freeway corridor. Therefore, these continuous variables were transformed into ordinal variables (as they appear in Table 1). To create these ordinal variables the original continuous variables were recursively split into groups until the association of the resultant grouping with the binary target *y* (representing crash or noncrash) is maximized. After variable selection the data set was partitioned into training (70%) and validation (30%) sets. Stratification with respect to target *y* was used so as to maintain same ratio of crash and noncrash cases in training and validation sets.

In the data set used for "training" the PNN observations belonging to the crash category were about 15% of the sample. Due to significantly more noncrash cases the number of neurons in the pattern layer would be large, which would increase the computation time during the application stage. To limit the number of non-crash cases, it was decided to use a balanced sample of crash and noncrash cases. One idea was to randomly select noncrash observations equal to the 15% crashes out of the completely random noncrash sample and use them for training along with the crash data points. The problem with this approach was that we would lose key contributions from a lot of available noncrash data points.

Hence, it was decided to reduce the observations belonging to noncrash cases by means of a clustering procedure. A subtractive clustering procedure was used to reduce noncrash observations (in the training data set) into cluster centers such that the number of noncrash cluster centers is equal to the number of crashes in the training data set. The subtractive clustering involves selecting an appropriate cluster radius such that the desirable number of noncrash cases is selected as cluster centers representing all noncrash observations lying within that particular radius. In the present application the desirable number of noncrash cases is equal to the crashes available in the training data set. These cases would be clustered out of all available training non-crash cases. It should be noted, however, that the non-crash data points in the validation data set were not clustered and were used as is to evaluate the performance of the models. The application of the subtractive clustering procedure at the "training" stage of PNN was proposed earlier by Abdel-Aty and Pande (2005). The clustering procedure limits the number of units in the pattern layer making potential real-time application computationally less cumbersome.

[Insert Table 1]

The critical parameter for a PNN is σ representing the spread parameter. For small values of the spread parameter the PNN reduces to the nearest neighbor classifier with each individual case exerting too much influence on the performance of the network. Higher values of σ cause the PNN to lose the details of density functions being estimated. The range examined to search for the optimal spread parameter was from 0.001 through 0.1 with an increment of 0.001. It essentially means that within each of the three sets (including traffic parameters from 1, 3, or 5 stations) 100 PNN models were estimated with varying values of the spread parameter.

These models were then applied to the validation data set. The output of these models (for any observation) is the posterior probability of the event of interest (i.e., a rear-end crash), which is a number between 0 and 1. According to the model, the observations with higher output posterior probability are more likely to be a rear-end crash compared to an observation with lower posterior probability.

For evaluating the performance of any PNN model the validation data set observations were sorted by the output posterior probability. In the sorted group, the top 10% observations would be the 10% observations that are most likely to be a crash. The performance of a model may be measured by determining the proportion of crashes in the validation data set captured within various deciles of posterior probability. Decile is defined as any of nine points that divide a distribution of ranked scores into 10 equal intervals with each interval containing one-tenth of the scores. As these models are intended to identify an event as rare as crashes to choose among competing models the proportion of crashes captured within the first few deciles must be critically examined. Also, note that applying a lower threshold to the output from the same model (to separate crash from noncrash cases) would lead to more positive decisions (i.e., crash classification) and hence would increase the false alarm rate. It was decided that the best model among a set of competing models would be the one capturing the highest percentage of validation data set crashes within the first three deciles (i.e., 30 percentile). The spread parameter value yielding the highest percentage of crashes in the top 30% observations was selected as the optimal value. It must be acknowledged that a 30 percentile threshold is chosen for demonstrating the results from the models calibrated herein and is not supposed to be a recommended threshold for real-time application.

Using this criterion the models with optimal value of the spread parameter σ were identified in each of the three sets (i.e., with input traffic parameters from 1, 3, or 5 stations). The percentage of crashes identified within the first three deciles of the output posterior probability along with the optimal spread parameter is provided in Table 2. The classification performance of the models also needs to be seen in relation to the performance of a "model" that randomly assigns observations as crash or noncrash. Essentially if 30% of observations are randomly assigned as crashes then 30% of crashes in the sample would be identified correctly. On the other hand, if classification (for observations in the validation data set) is assigned such that 30% of observations with highest posterior probability (i.e., model output) are designated as crashes, it results in identification of more than 50% of Regime 2 rear-end crashes. This difference between random assignment and classification based on the model(s) is a representative of the model's classification accuracy. This measure is also shown in Table 2 in parentheses. The improvement achieved by the models over random assignment signifies that the outputs from these models would in fact serve a measure of rear-end crash risk. Table 2 also lists the approximate percentage of false alarms in the validation data set. Note that the proportion of false alarms in a real-time application scenario would be very close to the proportion of positive decisions due to the rarity of crashes. These two measures (false alarm rate and improvement over random assignment) provided in Table 2 indicate that although these models may not be able to "predict" each and every Regime 2 rear-end crash, they can reliably assess traffic conditions for their crash potential.

[Insert Table 2]

In the next step, all combinations (i.e., the ensemble models) of the best PNN models (shown in Table 2) were created by averaging the posterior probabilities estimates for each observation in the validation data set from the individual models. The ensemble models combine information from multiple classification models. Note that the ensemble models used in this study are different than the ensemble models created by bagging and boosting methods. Bagging and boosting are used to combine information from models developed using different training data samples (Polikar, 2006).

For a binary target, an ensemble of multiple PNN models may alternatively be achieved by classifying the cases into the classes assigned to them by the majority of individual models. This method is called voting and is not equivalent to averaging posterior probabilities. Although voting could provide a predicted target value, it would not produce posterior probability estimates consistent with the individual posteriors. When an individual classifier assigns an output class label, the decision is based on a predetermined threshold. If the estimated posterior probability is less than this threshold then the classifier would produce 0 indicating noncrash, otherwise it would return a value of unity to indicate a crash. The output of an ensemble classifier, according to the voting method, would be based on the majority of class labels from multiple classifiers. Observations assigned as crashes according to the "majority-rule" ensemble

classifier cannot be compared among each other. In other words, there would be no way to judge which pattern is more crash-prone among all the patterns that are identified as potential crashes. However, if the ensemble model is estimated by averaging the posterior probabilities, it is still possible to rank the observations in the validation data set to create lift plots. It will in turn help in evaluating the performance of the ensemble model vis-`a-vis the individual models.

The lift plots depicting the performance of all possible ensemble models are shown in Figure 3. The plot shows the percentage of the lane-change crashes in the validation data set captured within various deciles of posterior probability by each model on the Y-axis. On the X-axis the percentiles are shown at equal intervals of 10. Models may be assessed by examining the value of their ordinate, with higher ordinate(s) indicating better model(s). The curve shown as *best135* (combination of three, i.e., best 1-station, 3-station, and 5 station PNN models from Table 2) runs higher than other lift curves in the vicinity of abscissa value of 30 percentile. It is slightly above the curve belonging to the ensemble model titled *best35* (representing the combination of best 3-station and 5-station PNN models), *best15* (representing the combination of best 1-station and 5-station PNN models), and *best13* (representing the combination of best 1-station and 3-station PNN models). At 30 percentile the combination of the three models (i.e., best135) captures the maximum percentage of crashes (57.89% of the crashes from the validation sample) and is, therefore, recommended for identification of Regime 2 rear-end crashes. In comparison, the best individual PNN model using data from five stations (see last row in Table 2) only identified 53.20% of crashes. Hence, a sizeable improvement in crash identification was achieved through the ensemble models created by averaging the outputs of the individual models.

It is worth mentioning at this point that this performance was achieved through the ensemble model, which would use traffic inputs from five stations (Station D through H). As data from five stations may not be simultaneously available due to intermittent failures of loops, performance of the models must be seen in terms of their data requirements as well. Sometimes it may be more practical to use data from one or three stations to identify these crashes. Therefore, even though the 5 station model provides better identification of Regime 2 crashes, it would not make it an automatic choice for field implementation.
[Insert Figure 3]

5.3 Implementation strategy

A preliminary field implementation plan may be formulated based on the discussion provided so far. If the traffic conditions are identified as Regime 1, the data pattern may be declared as potentially crash prone and warning (or other mitigation strategy) for a rear-end crash can be issued. If the data are found to be associated with Regime 2 traffic speed conditions, they may be subjected to the ensemble *PNN* model(s) developed in the previous section. Note that the PNN models are designed to separate crashes within the data satisfying Regime 2 traffic conditions.

With this strategy one can expect to identify 46% of rear-end crashes (percentage of Regime 1 crashes among all rear-end crashes) by issuing warnings for about 7% of cases. 57.89% of Regime 2 crashes, which make up 54% of the rear-end crash data, may be identified by issuing warnings 30% of the time among the remaining 93% of cases. It essentially means that about three-fourth (46 + (54 x 58)/100 ~ 77%) of the crashes could be identified by issuing warnings for about one-third of the cases (7 + (93 x 30)/100 ~ 34%). It roughly translates into 66% accuracy on noncrash data for identification of 77% of crashes. It is worth mentioning that as crashes are rare events, the “false-alarm” rate during real-time application would be very close to the proportion of positive decisions. This classification accuracy is more than that achieved by any generic models in the past (e.g., Abdel-Aty et al., 2004). The percentile threshold for identification of Regime 2 rear-end crashes may be varied to achieve a more desirable balance between positive decisions and proportion of identified crashes.

In one of our recent studies we discussed the real-time application framework for identification of crash-prone conditions (Pande and Abdel-Aty, 2007). It was noticed that even with the reduced number of neurons in the pattern layer (deduced using a subtractive clustering

procedure), the computational time for PNN remained higher compared to multilayer perceptron (MLP) networks with comparable performance. It might be a constraint for applying these models in a real-time application framework. The procedure for developing NRBF (normalized radial basis function) and MLP (multilayer perceptron) networks may be found in Pande and Abdel-Aty (2006).

6 CONCLUSIONS

This study presents a step-by-step approach of data analysis to develop a preliminary strategy for identify-
ing traffic conditions prone to rear-end crashes in an ATMS (advanced traffic management system) framework. It was concluded that the rear-end crashes on the freeway may be grouped into two distinct clusters (traffic regimes) based on the average speeds prevailing in an approximately 2-mile section around the crash location 5–10 minutes before a crash. Essentially, Regime 1 crashes are the ones that occur when the congested conditions have already set in and could be observed at loop detector(s) at least 5–10 minutes before the crash. For Regime 2 rear-end crashes the traffic conditions at crash locations were relatively less-congested 5–10 minutes before the crash.

A preliminary real-time application strategy involving the classification tree-based rules and ensemble PNN model(s) was proposed. It was shown that using the proposed strategy, more than 77% of the rear-end crashes may be identified at least 5 minutes before their occurrence with about 34% positive decisions (i.e., crash warnings). As crashes (however, frequent on the I–4 corridor under consideration) are such rare events, 34% positive decisions would result in a "false-alarm" rate of only slightly less than 34%. Even if one may bring down the false alarm rate using a higher threshold for the identification of Regime 2 rear-end crashes, it would still remain significant. However, it should be noted that "false alarms" are not as detrimental for the present application as they would be for incident detection algorithms.

It is worth mentioning that while the application and solution approach seems similar to incident detection, the performance metric for the model performance is not as simple as the false alarm rate. In fact, the ultimate goal of this research would, or at least should be, to "achieve" a "false alarm" every time a crash warning is issued. The goal would be based on the expectation that with some form of proactive countermeasure or warnings to the motorists, potential crashes following the crash-prone conditions may be avoided.

Even without the countermeasures it is neither improbable nor unacceptable to have these "false alarms." The outcome in case of proactive assessment of crash risk is not as simple as the postfacto detection of incident. The idea here is to develop a framework based on historical crashes that can assess future traffic conditions on the freeway. Crash-prone traffic conditions identified based on the methodology proposed herein would not always result in a rear-end crash occurrence even though a significant proportion of historical crashes did occur under those conditions. Hence, drivers need to be more attentive under such traffic conditions even if they may not always culminate in a rear-end crash. The nature of warning, for example, would not be "crash WILL occur" but more measured such as "drive carefully" or "watch for queues." A reasonable number of

warnings that the drivers do not consider excessive, may play a critical role in (proactive) traffic management.

The nature and optimal number of warnings issued to the motorists (so as to keep the warnings effective) would require relating this work with human factors research.

ACKNOWLEDGMENT

The authors wish to thank the Florida Department of Transportation for funding this research. Crash data are obtained from the FDOT database and loop data from the UCF/CATSS data warehouse. All opinions and results are those of the authors.

REFERENCES

Abdel-Aty, M. & Pande, A. (2005), Identifying crash propensity using specific traffic speed conditions, *Journal of Safety Research*, **36**, 97–108.

Abdel-Aty, M., Uddin, N., Abdalla, F., Pande, A. & Hsia, L. (2004), Predicting freeway crashes based on loop detector data using matched case-control logistic regression, *Transportation Research Record 1897*, pp. 88–95.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Chapman and Hall, New York.

Davis, J. & Goadrich, M. (2006), The relationship between precision-recall and ROC curves, in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA.

Golob, T. F. & Recker, W. W. (2001), Relationships among urban freeway accidents, traffic flow, weather and lighting conditions, California PATH Working Paper UCB-ITSPWP-2001-19, Institute of Transportation Studies, University of California, Berkeley, CA.

Golob, T. F., Recker, W. W. & Alvarez, V. M. (2004), Freeway safety as a function of traffic flow, *Accident Analysis & Prevention*, **36**(6), 933–46.

Lee, C., Hellinga, B. & Saccomanno, F. (2003), Real-time crash prediction model for the application to crash pre-
vention in freeway traffic, *Transportation Research Record 1840*, pp. 67–78.

Lee, C., Saccomanno, F. & Hellinga, B. (2002), Analysis of crash precursors on instrumented freeways, *Transportation Research Record 1784*, pp. 1–8.

Ma, K. & Morgan, N. (1995), Scaling down: applying large vocabulary hybrid HMM-MLP methods to telephone recognition of digits and natural numbers, in *Proceedings of the 5th IEEE Workshop on Neural Networks for Signal Processing*, Cambridge, MA.

Madanat, S. & Liu, P. (1995), A prototype system for real-time incident likelihood prediction, IDEA Project Final Report (ITS-2), Transportation Research Board, National Research Council, Washington, DC.

Masters, T. (1995), *Advanced Algorithms for Neural Networks: A C++ Sourcebook*, John Wiley and Sons, New York.

Oh, C., Oh, J., Ritchie, S. & Chang, M. (2001), Real-time estimation of freeway accident likelihood, presented at the 80th annual meeting of the Transportation Research Board, Washington, DC.

Pande, A. (2005), Estimation of hybrid models for real-time crash risk assessment on freeways, Ph.D. Dissertation, University of Central Florida, Orlando, FL.

Pande, A. & Abdel-Aty, M. (2006), A comprehensive analysis of the relationship between real-time traffic surveillance data and rear-end crashes on freeways, *Transportation Research Record 1953*, pp. 31–40.

Pande, A. & Abdel-Aty, M. (2007), A multiple-model framework for real-time crash risk assessment, *Transportation Research Record 2019*, pp. 99–107.

Pande, A., Abdel-Aty, M. & Hsia, L. (2005), Spatio-temporal variation of risk preceding crash occurrence on freeways, *Transportation Research Record 1908*, pp. 26–36.

Polikar, R. (2006), Ensemble based systems in decision making, *IEEE Circuits and Systems Magazine*, **6**(3), 21–45.

Provost, F., Fawcett, T. & Kohavi, R. (1998), The case against accuracy estimation for comparing induction algorithms, in *Proceeding of the 15th International Conference on Machine Learning*, San Francisco, CA.

Specht, D. F. (1990), Probabilistic neural networks, *Neural Networks*, **3**, 110–18.

Specht, D. F. (1996), Probabilistic neural networks and general regression neural networks, in C. H. Chen (ed.), *Fuzzy Logic and Neural Network Handbook*, McGraw-Hill, Berlin, pp. 3.1–3.37.

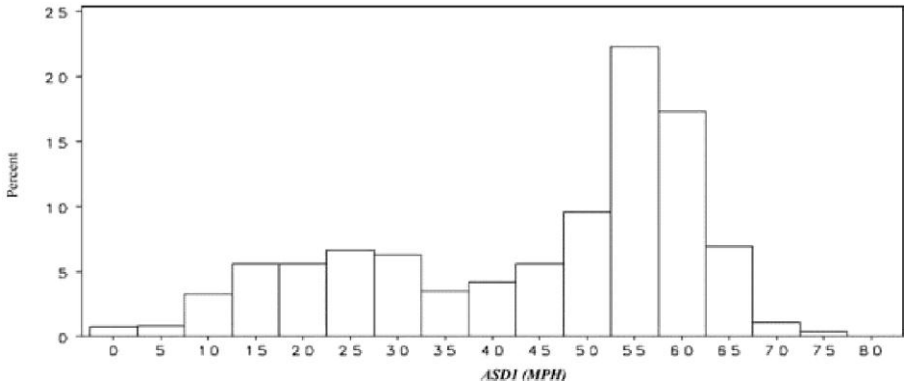

Fig. 1. Histogram distribution of 5-minute average speeds at Station D before all rear-end crashes.

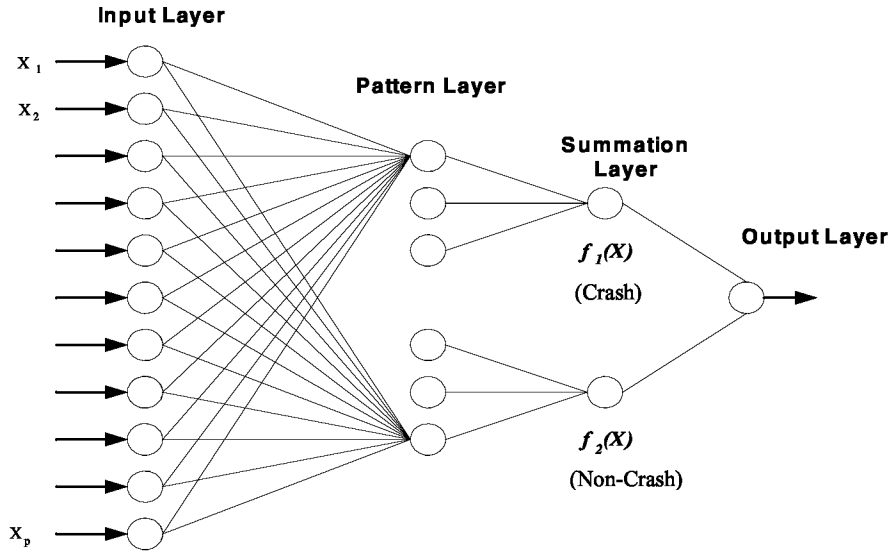

Fig. 2. The PNN architecture for a two-class classification problem.

Table 1
Variables used as input to the PNN for the three stages

List of traffic factors selected through tree model with		
Traffic parameters only from station F	*Traffic parameters from stations E, F, and G*	*Traffic parameters from stations D, E, F, G, and H*
ASF2, AVF2, SOF2, AOF2	*ASG2, ASF2, AOF2, AVF2, SSG2, SOG2*	*ASG2, ASF2, ASH2, AOF2, SOG2*

List of off-line factors selected through tree model with		
Traffic parameters only from station F	*Traffic parameters from stations E, F, and G*	*Traffic parameters from stations D, E, F, G, and H*
DOWNSTREAMOFF = 0 if nearest downstream off-ramp is located further than 0.0638 miles = 1 if nearest downstream off-ramp is located within 0.0638 miles	*DOWNSTREAMON* = 0 if nearest downstream on-ramp is located further than 0.7747 miles = 1 if nearest downstream on-ramp is located within 0.7747 miles	*CRASHTIME* = 0 if time of crash between midnight and 12:26 AM = 1 if time of crash between 12:26 AM and 6:46 AM = 2 if time of crash between 6:46 AM and 7:24 PM = 3 if time of crash between 7:24 PM and midnight
DOWNSTREAMON = 0 if nearest downstream on-ramp is located further than 0.7747 miles = 1 if nearest downstream on-ramp is located within 0.7747 miles	*DOWNSTREAMOFF* = 0 if nearest downstream off-ramp is located further than 0.0638 miles = 1 if nearest downstream off-ramp is located within 0.0638 miles	*DOWNSTREAMON* = 0 if nearest downstream on-ramp is located further than 0.7747 miles = 1 if nearest downstream on-ramp is located within 0.7747 miles
BASE_MILEPOST = 0 if 0<base_milepost<=11.93 = 1 if 11.93<base_milepost<=25.43 = 2 if 25.43<base_milepost<=35.18 = 3 if 35.18<base_milepost<=36.25	*CRASHTIME* = 0 if time of crash between midnight and 12:26 AM = 1 if time of crash between 12:26 AM and 6:46 AM = 2 if time of crash between 6:46 AM and 7:24 PM = 3 if time of crash between 7:24 PM and midnight	*UPSTREAMOFF* = 0 if nearest upstream off-ramp is located further than 0.3205 miles = 1 if nearest upstream off-ramp is located within 0.3205 miles
STATION F = 0 if loop detector station nearest to crash location is located upstream = 1 if loop detector station nearest to crash location is located downstream	*UPSTREAMOFF* = 0 if nearest upstream off-ramp is located further than 0.3205 miles = 1 if nearest upstream off-ramp is located within 0.3205 miles	*BASE_MILEPOST* = 0 if 0<base_milepost<=11.93 = 1 if 11.93<base_milepost<=25.43 = 2 if 25.43<base_milepost<=35.18 = 3 if 35.11<base_milepost<=36.25
CRASHTIME = 0 if time of crash between midnight and 12:26 AM = 1 if time of crash between 12:26 AM and 6:46 AM = 2 if time of crash between 6:46 AM and 7:24 PM = 3 if time of crash between 7:24 PM and midnight	*BASE_MILEPOST* = 0 if 0<base_milepost<=11.93 = 1 if 11.93<base_milepost<=25.43 = 2 if 25.43<base_milepost<=35.18 = 3 if 35.18<base_milepost<=36.25	

Table 2
Percentage of Regime 2 rear-end crashes captured along with the corresponding false alarms for the best models at each of the three stages

	Percentage of crashes identified within 30 percentile of posterior probability (a) (b**)*	*Approximate percentage of false alarms (30–15a)*
Traffic parameters from		
Station F	49.21% (19.21%*) ($\sigma = 0.041$**)	22.61%
Station E, F, and G	52.90% (22.90%*) ($\sigma = 0.060$**)	22.06%
Station D, E, F, G, and H	53.20% (23.20%*) ($\sigma = 0.083$**)	22.02%

*a = Percentage of additional crashes identified compared to a random assignment scheme.
**b = Optimal spread parameter.

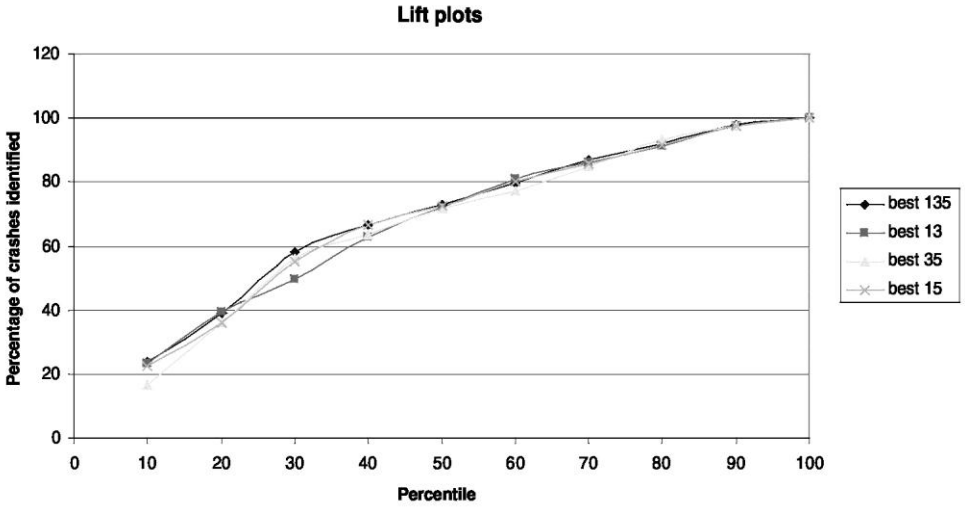

Fig. 3. Percentage of captured response lift plot for all possible ensemble models for Regime 2 rear-end crashes.